

oooo

LỚP L01 - NHÓM 5

# KHO DỮ LIỆU VÀ HỆ HỖ TRỢ QUYẾT ĐỊNH

oooo<sub>1</sub>

# **TÊN ĐỀ TÀI**

XÂY DỰNG DATA WAREHOUSE TRÊN  
NỀN TẢNG NGHE NHẠC SPARKIFY

## **THÀNH VIÊN**

Nguyễn Đức Bình  
MSSV: 2112899

Lâm Tấn Thịnh  
MSSV: 2110559

Nguyễn Duy Tùng  
MSSV: 2115232

# NỘI DUNG

- Giới thiệu
- Phân tích dữ liệu
- Kiến trúc
- Dự đoán trình duyệt
- Dự đoán nghệ sĩ



0 0 0 0

# GIỚI THIỆU

Trong thời đại số hóa, các nền tảng nghe nhạc trực tuyến như Sparkify tạo ra một lượng lớn dữ liệu từ hành vi người dùng, danh sách bài hát, lượt nghe, v.v. Tuy nhiên, dữ liệu này thường được lưu trữ dưới nhiều định dạng khác nhau, gây khó khăn cho việc phân tích. Để giải quyết vấn đề, nhóm xây dựng Kho dữ liệu (Data Warehouse) giúp tổng hợp, tổ chức và phân tích dữ liệu hiệu quả hơn.

Dự án tập trung vào:

- Trích xuất dữ liệu từ Amazon S3 (JSON).
- Tải vào bảng trung gian (staging tables).
- Chuyển đổi sang lược đồ bông tuyết (Snowflake Schema).
- Xây dựng các bảng tối ưu hóa cho phân tích dữ liệu.

# MÔ TẢ DỮ LIỆU

Dữ liệu chính được lấy từ Amazon S3, bao gồm:

- **Dữ liệu bài hát (Song Data):** Metadata về bài hát, nghệ sĩ (lấy từ Million Song Dataset).
- **Dữ liệu nhật ký (Log Data):** Nhật ký hoạt động của người dùng, ghi nhận lượt phát nhạc, thông tin thiết bị, trình duyệt,...

Nhóm xây dựng kho dữ liệu với 5 giai đoạn:

1. **Data Source:** Dữ liệu gốc từ JSON.
2. **Staging Area:** Chuyển dữ liệu thành bảng để dễ truy cập.
3. **Data Warehouse:** Tổ chức dữ liệu theo lược đồ bông tuyết.
4. **Data Mart:** Trích xuất dữ liệu phục vụ phân tích và dự đoán.
5. **Usage:** Áp dụng mô hình học máy để phân tích và dự đoán.

# **HƯỚNG PHÂN TÍCH VỚI TẬP DỮ LIỆU**

## **1. Hành vi khách hàng**

- Người dùng nghe nhạc vào khoảng thời gian nào trong ngày?
- Xu hướng nghe nhạc theo khu vực?
- Người dùng truy cập bằng hệ điều hành và trình duyệt nào?
- Tỉ lệ người dùng sẵn sàng trả phí để nghe nhạc?

## **2. Ảnh hưởng của bài hát và nghệ sĩ**

- Nghệ sĩ nào có nhiều người nghe nhất?
- Bài hát nào có lượng người nghe cao nhất?
- Bài hát nào phổ biến nhất theo khu vực?

# **HƯỚNG DỰ ĐOÁN VỚI TẬP DỮ LIỆU**

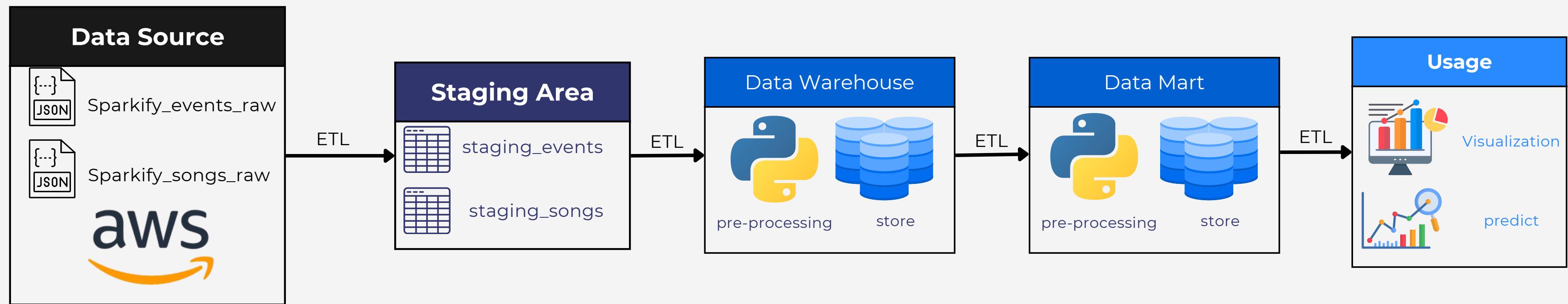
## **1. Dự đoán trình duyệt phổ biến của người dùng trong tương lai?**

- Xác định trình duyệt mà người dùng sẽ sử dụng để nghe nhạc dựa trên các đặc điểm như vị trí, thời gian nghe, nghệ sĩ yêu thích, độ dài bài hát, v.v.

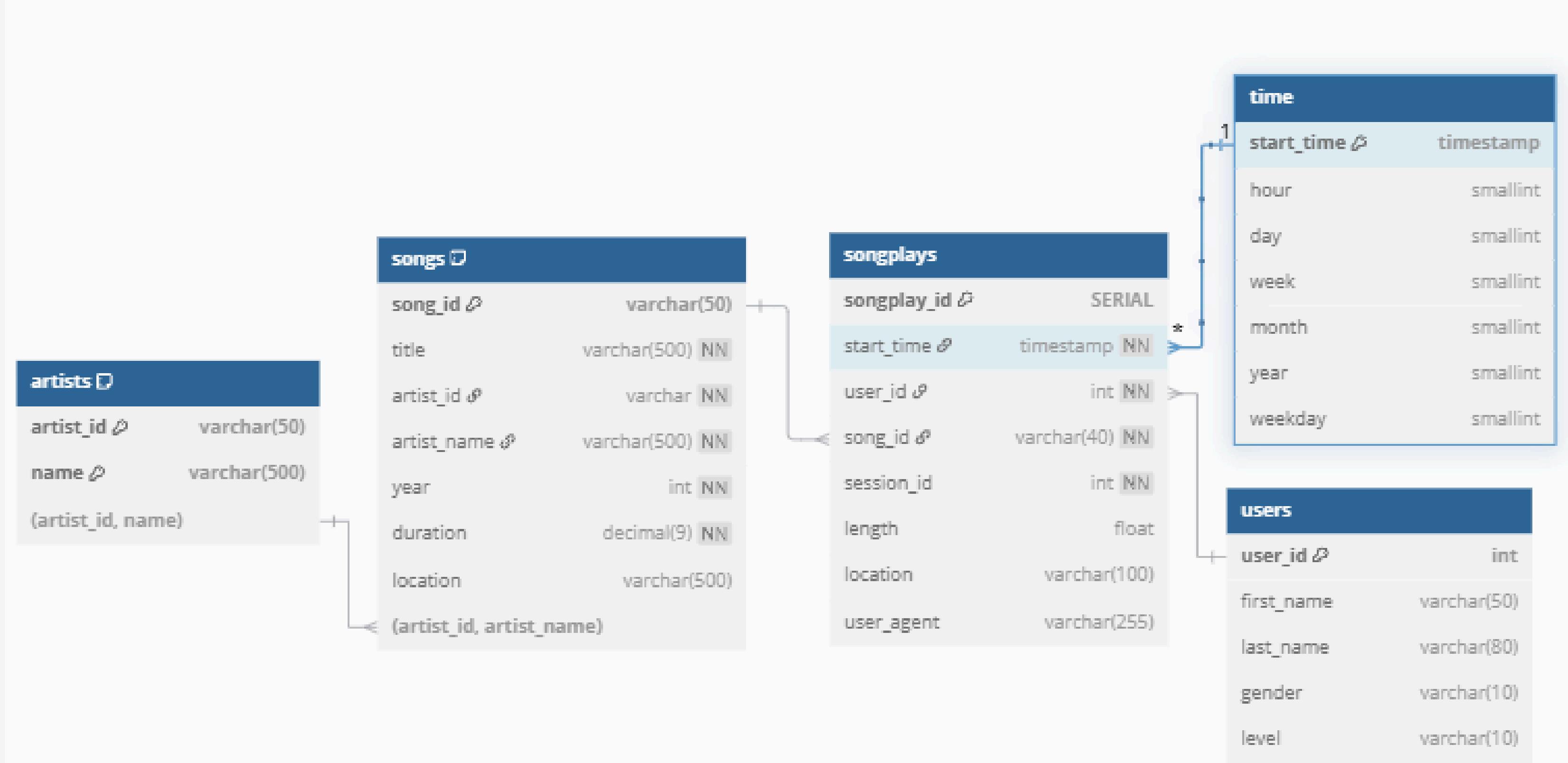
## **2. Dự đoán nghệ sĩ yêu thích của người dùng**

- Dự đoán nghệ sĩ mà người dùng có khả năng nghe tiếp theo dựa trên hành vi nghe nhạc trước đó.

# KIẾN TRÚC TỔNG QUAN

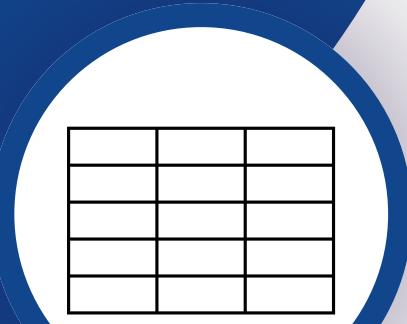


# LƯỢC ĐỒ BÔNG TUYẾT

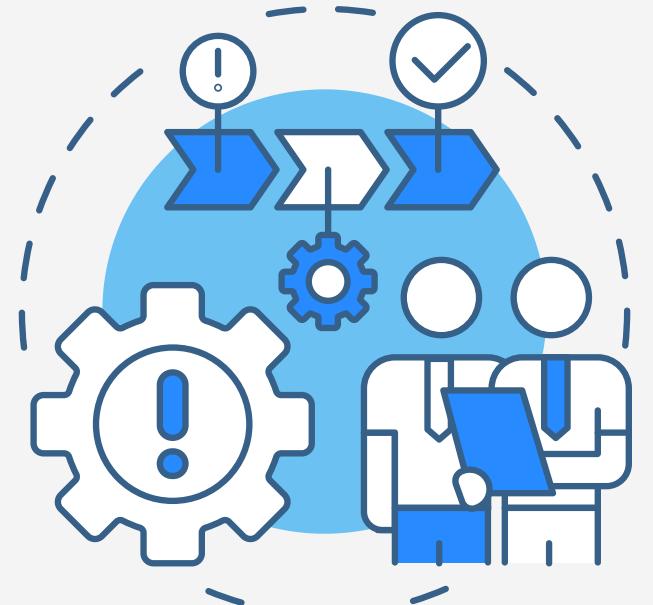


# ETL

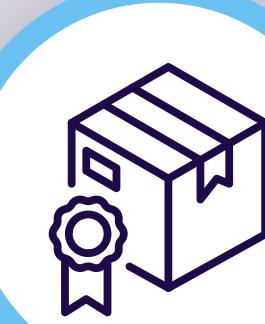
Thêm dữ liệu  
staging



Tạo bảng

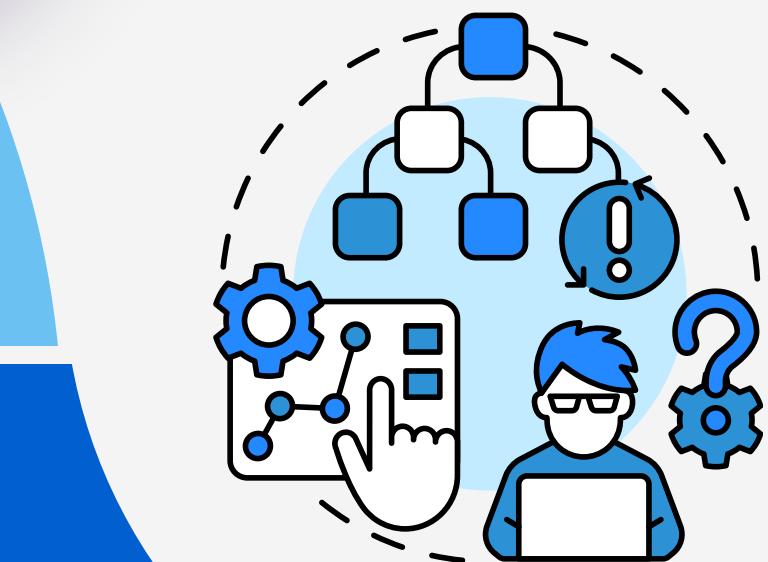


Tiền xử lí data  
warehouse



Tiền xử lí và nạp dữ liệu  
vào data mart

Nạp dữ liệu vào file .csv



# DATA SOURCE

- artist\_id: Mã định danh nghệ sĩ
- artist\_location: Vị trí của nghệ sĩ
- artist\_name: Tên nghệ sĩ
- duration: Thời lượng bài hát (tính bằng giây)
- song\_id: Mã định danh bài hát
- title: Tiêu đề bài hát
- artist\_latitude: Vĩ độ của nghệ sĩ
- artist\_longitude: Kinh độ của nghệ sĩ
- year: Năm phát hành (0 nếu không xác định)
- num\_songs: Số lượng bài hát

## Song data

- Một số vấn đề của bộ dữ liệu ???

- artist: Tên nghệ sĩ
- auth: Trạng thái xác thực người dùng (vd: 'Logged In', 'Logged Out')
- firstName: Tên của người dùng
- gender: Giới tính của người dùng
- itemInSession: Thứ tự của sự kiện trong một phiên
- lastName: Họ của người dùng
- length: Thời lượng bài hát (nếu sự kiện liên quan đến bài hát)
- level: Cấp độ tài khoản người dùng (vd: 'free', 'paid')
- location: Vị trí của người dùng
- page: Trang/Hành động người dùng truy cập (vd: 'Login', 'NextSong')
- registration: Thời điểm đăng ký của người dùng
- sessionId: Mã định danh phiên làm việc
- song: Tiêu đề bài hát (nếu sự kiện liên quan đến bài hát)
- ts: Dấu thời gian (timestamp) của sự kiện (thường là mili giây epoch)
- userId: Mã định danh người dùng
- method: Phương thức HTTP (vd: 'GET', 'PUT')
- userAgent: Thông tin trình duyệt/thiết bị của người dùng
- status: Mã trạng thái HTTP (vd: 200, 307)

## Event data

# TẠO BẢNG

## Mục Đích

```
# QUERY LISTS
create_table_queries = [
    staging_events_table_create,
    staging_songs_table_create,
    user_table_create,
    artist_table_create,
    song_table_create,
    time_table_create,
    songplay_table_create
]
```

- Tạo các bảng **staging** và **data warehouse**
- Xác định các cột, kiểu dữ liệu, và ràng buộc (ví dụ: khóa chính, khóa ngoại).
- Các bảng cần tạo:
  - Staging\_events,
  - Staging\_songs,
  - Songplays,
  - Users,
  - Songs,
  - Artists,
  - Times

# DATA STAGING

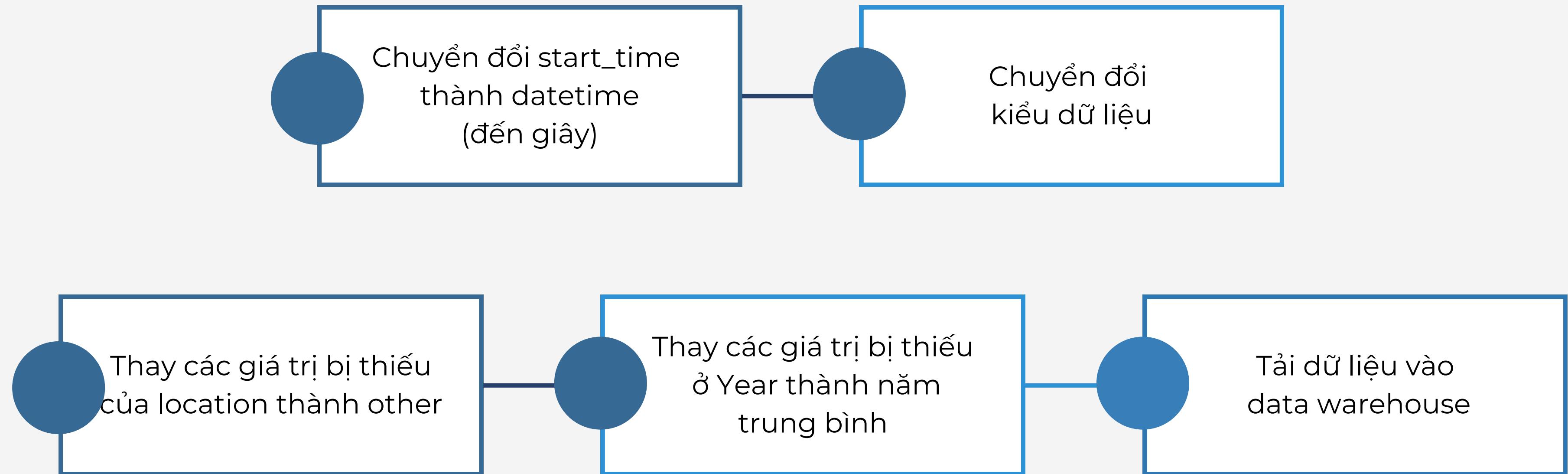
* event_id	artist	auth	firstname	gender	iteminsession	lastname	length	level	location	method
integer	varchar	varchar	varchar	varchar	varchar	varchar	varchar	varchar	varchar	varchar
4	(NULL)	Logged In	Kaylee	F	2	Summers	(NULL)	free	Phoenix-Mesa-Scottsdale, AZ	GET
5	Mr Oizo	Logged In	Kaylee	F	3	Summers	144.03873	free	Phoenix-Mesa-Scottsdale, AZ	PUT
6	Tamba Trio	Logged In	Kaylee	F	4	Summers	177.18812	free	Phoenix-Mesa-Scottsdale, AZ	PUT
7	The Mars Volta	Logged In	Kaylee	F	5	Summers	380.42077	free	Phoenix-Mesa-Scottsdale, AZ	PUT
8	Infected Mushroom	Logged In	Kaylee	F	6	Summers	440.2673	free	Phoenix-Mesa-Scottsdale, AZ	PUT
9	Blue October / Imogen Hea	Logged In	Kaylee	F	7	Summers	241.3971	free	Phoenix-Mesa-Scottsdale, AZ	PUT
10	Girl Talk	Logged In	Kaylee	F	8	Summers	160.15628	free	Phoenix-Mesa-Scottsdale, AZ	PUT

## Event data

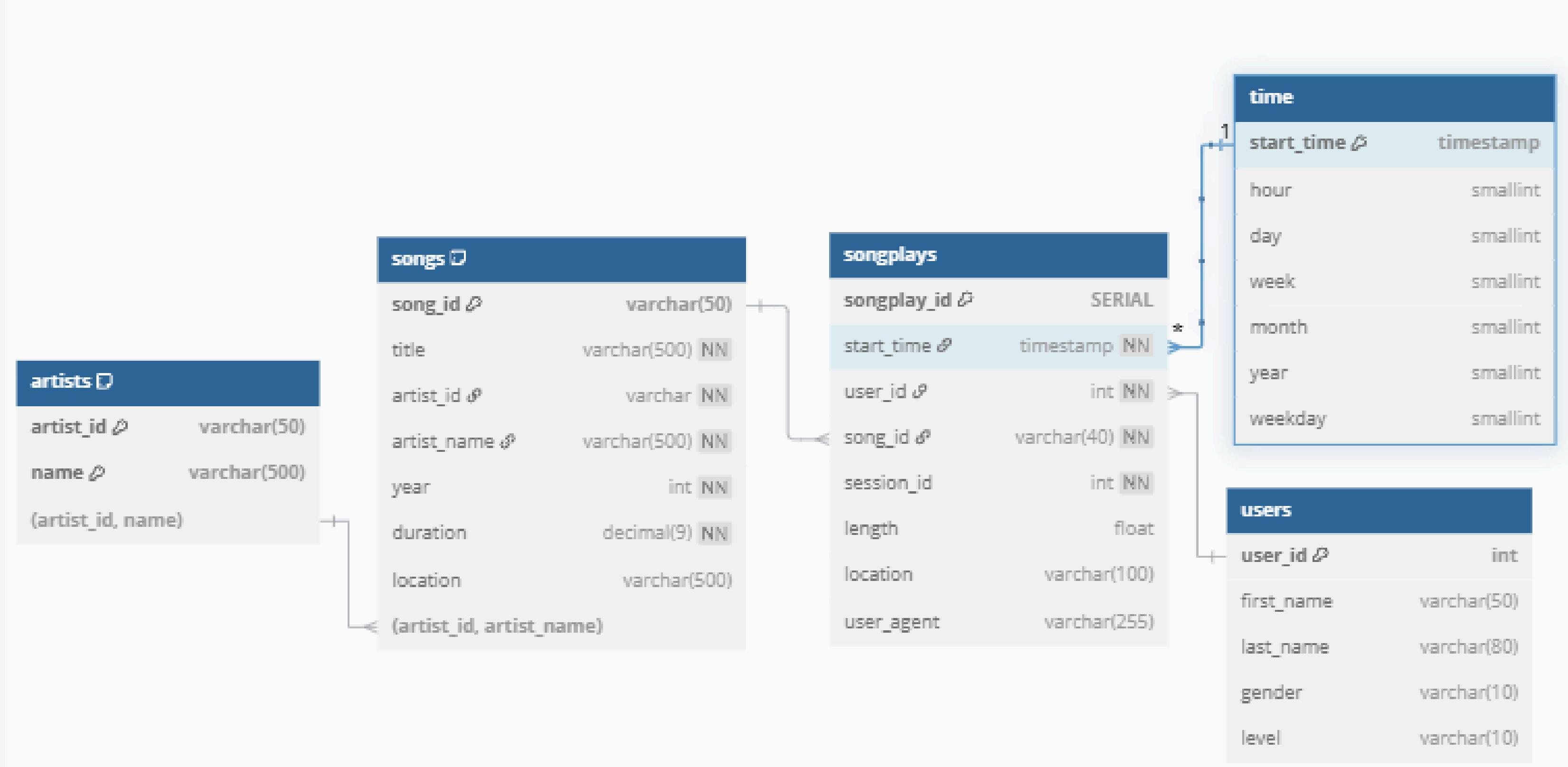
num_songs	* artist_id	artist_latitude	artist_longitude	artist_location	artist_name	* song_id	title	dura
integer	varchar	varchar	varchar	varchar(500)	varchar(500)	varchar	varchar(500)	num
1	AR7DQY51187B9B2D8E	31.1689	-100.07715	Texas	Chingo Bling w/ Baby Bash	SOEQFHG12A81C204A0	Head Honcho (w/ Baby Bas	212
1	AR2I4Q41187B992493	(NULL)	(NULL)		Red Peters	SOWRJDI12AB018AFC9	Pullin' It All Night Long	33
1	AR3752L1187FB4B67E	(NULL)	(NULL)		The Real Kids	SOUYDOZ12A8C131C6	Happens All The Time	192
1	AR3SX9G1187FB52D72	35.21962	-80.01955	North Carolina	No Mercy	SOWHEAR12A8C13CE4	Please Don't Go	241
1	ARJ0EJP1187B9930E8	(NULL)	(NULL)		Loquillo	SOVJZYB12AB017B4F2	La mala reputacion	135
1	ARBEOHF1187B9B044D	(NULL)	(NULL)	Bay City, MI	Madonna	SOIYPLX12AB0189CA2	Holiday	368
1	ART8OXR1187B9A45C5	53.3667	-6.3	Cabra, Dublin, Ireland	Declan Masterson	SOOEPC12AB0180A9C	Fairy Child	216
1	ARBP1UW1187FB5830C	(NULL)	(NULL)		The Sticks	SOYOCQX12AB0187905	Earshot	91

## Song data

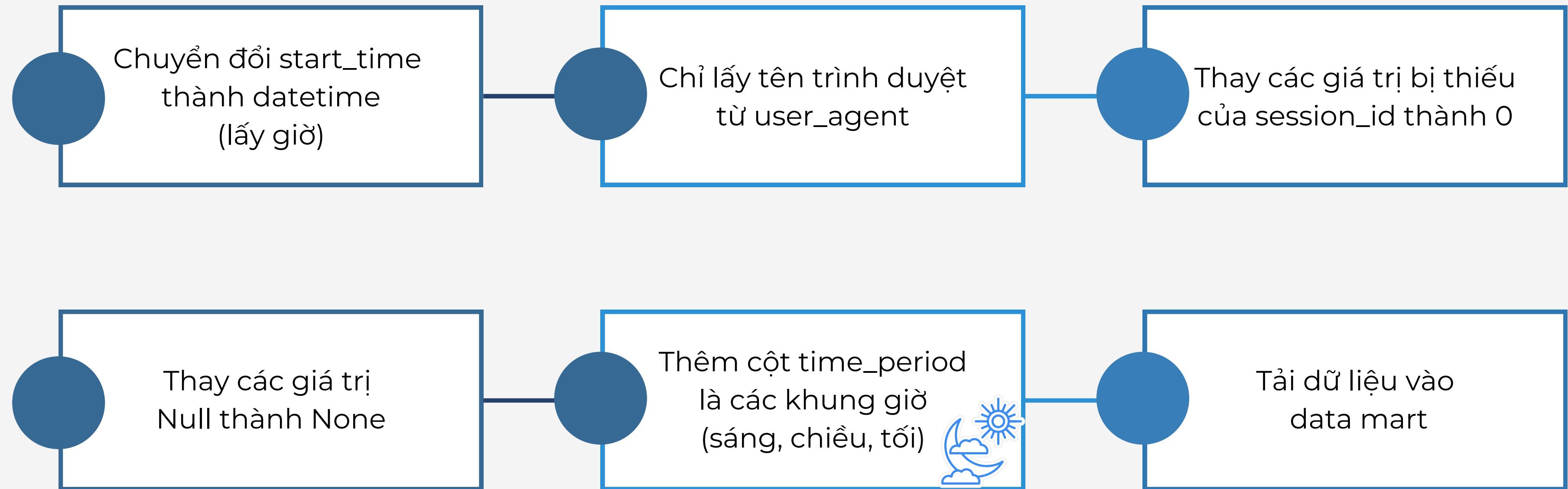
# TIỀN XỬ LÍ VÀ TẢI DỮ LIỆU DATA WAREHOUSE



# LƯỢC ĐỒ BÔNG TUYẾT



# TIỀN XỬ LÍ VÀ TẢI DỮ LIỆU VÀO DATA MART



# LOADING DATA TO CSV AND DATABASE

data_mart	
songplay_id	SERIAL
start_time	timestamp
user_id	int
song_id	varchar(40)
title	varchar(500)
artist_id	varchar(50)
artist_name	varchar(100)
session_id	int
length	float
location	varchar(255)
browser	varchar(255)
time_period	varchar(50)
level	varchar(10)

	A	B	C	D	E	F	G	H	I	J
1	songplay_id	start_time	user_id	song_id	title	artist_id	artist_nam	session_id	length	location
2		1 11/23/2018 15:00		30 SOHIBSG1	Eye Of The ART5MUE1	Metallica		691	415.1636	San Jose
3		2 11/23/2018 15:00		88 SOBVEAF1	Durch den ARVVHDT1	Tokio Hote		812	220.3424	Sacrame
4		3 11/29/2018 17:00		75 SOTUNMH	Nashville FARRJNTE1	Casiotone		721	139.4151	Columbi
5		4 11/2/2018 12:00		15 SOXNJYI12	Bite Back ARR1R5V1	The All-Am		172	239.2289	Chicago
6		5 11/4/2018 20:00		73 SOBGPAH	Heaven's II ARRJ3UC1	Wyclef Jea		72	242.0763	Tampa-S
7		6 11/9/2018 15:00		6 SOZKFHV1	I'm Not Mo AR9W3X91	Phil Collins		406	239.8559	Atlanta-S
8		7 11/5/2018 18:00		101 SOQJOUF1	Looks Like ARNZPSB1	NEEDTOBF		282	209.8673	New Orle
9		8 11/16/2018 21:00		49 SOHTWLT1	Inside Job ARFVYJI11	Pearl Jam		648	179.3302	San Fran
10		9 11/5/2018 12:00		44 SOTXVK1	déranger le ARSN9QH	Carla Brun		269	163.0298	Waterloo

**3.39K**

Count of Songs

**18.75%**

Percentage Paid Users

**546K**

Sum of Users

**1.73K**

Count of Artists

**Sum of users**

by Year, Quarter, Month and Day

40K

20K

0K

Nov 04

Nov 11

Nov 18

Nov 25

**Sum of users**

by location

Lansing-East Lansing, MI

Portland-South Portland...

San Francisco-Oakland-...

Sacramento--Roseville--...

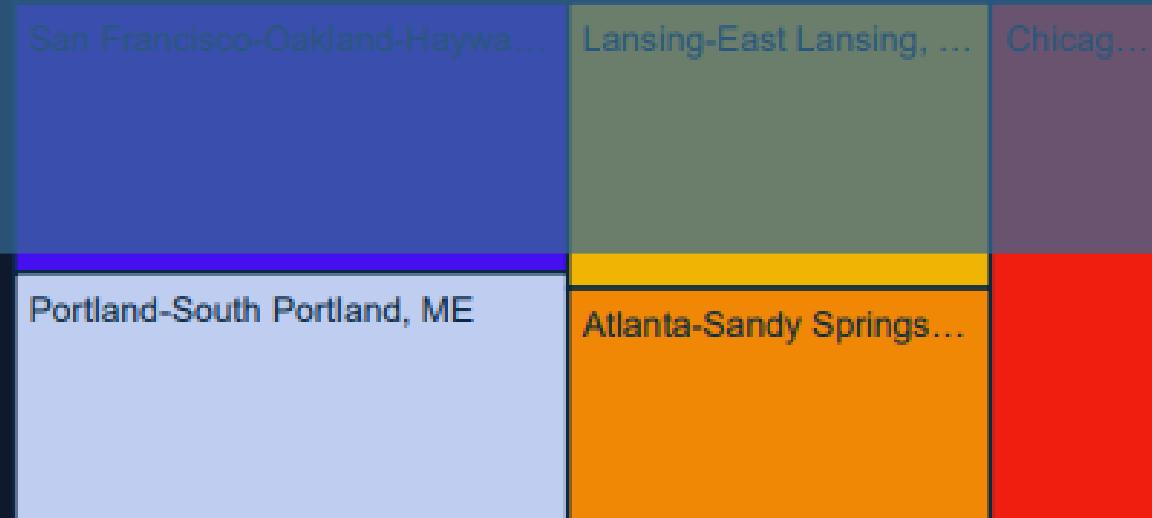
Tampa-St. Petersburg-C...

Atlanta-Sandy Springs-...

0K

50K

100K

**VISUALIZATION****Location**

- Atlanta-Sandy Springs-Roswell, GA
- Augusta-Richmond County, GA-SC
- Birmingham-Hoover, AL
- Cedar Rapids, IA
- Chicago-Naperville-Elgin, IL-IN-WI

**Count of songplays**

by browser

**5.8K**

4K

2K

0K

Chrome

**2.3K**

Firefox

**1.8K**

Safari

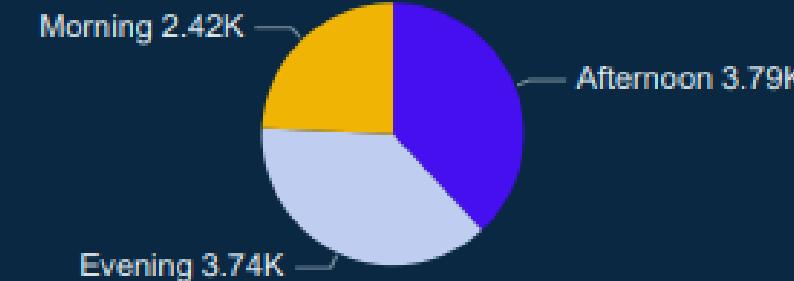
**0.1K**

Trident

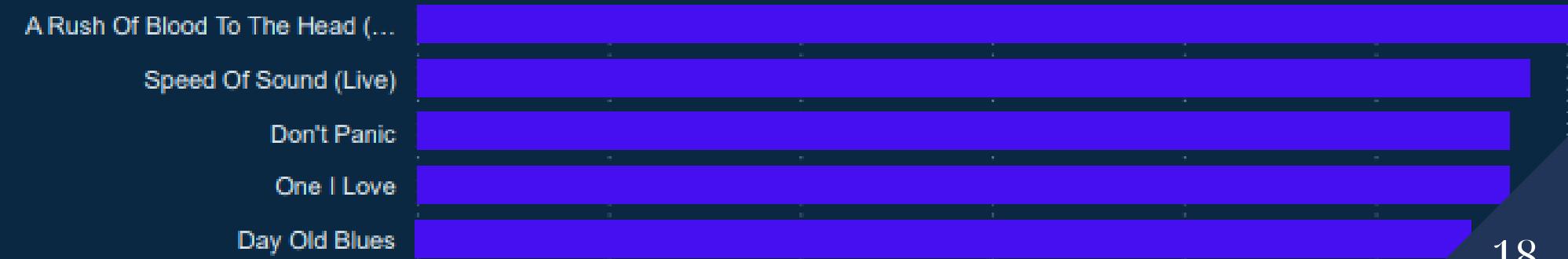
**Count of songplays**

by time\_period

Afternoon Evening Morning

**Count of songplays**

by title



18

**3.39K**

Count of Songs



**546K**

Sum of Users

**18.75%**

Percentage Paid Users



**1.73K**

Count of Artists



**Sum of user**  
by Year, Quarter, Month and Day

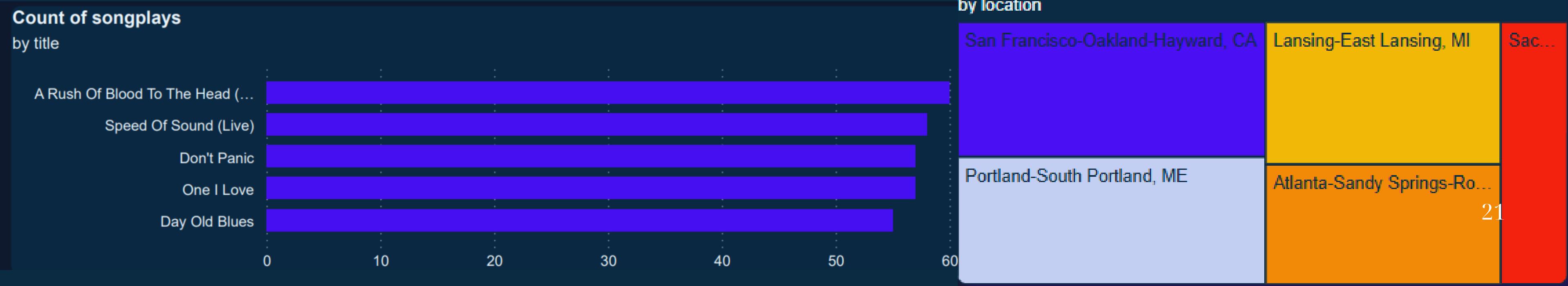
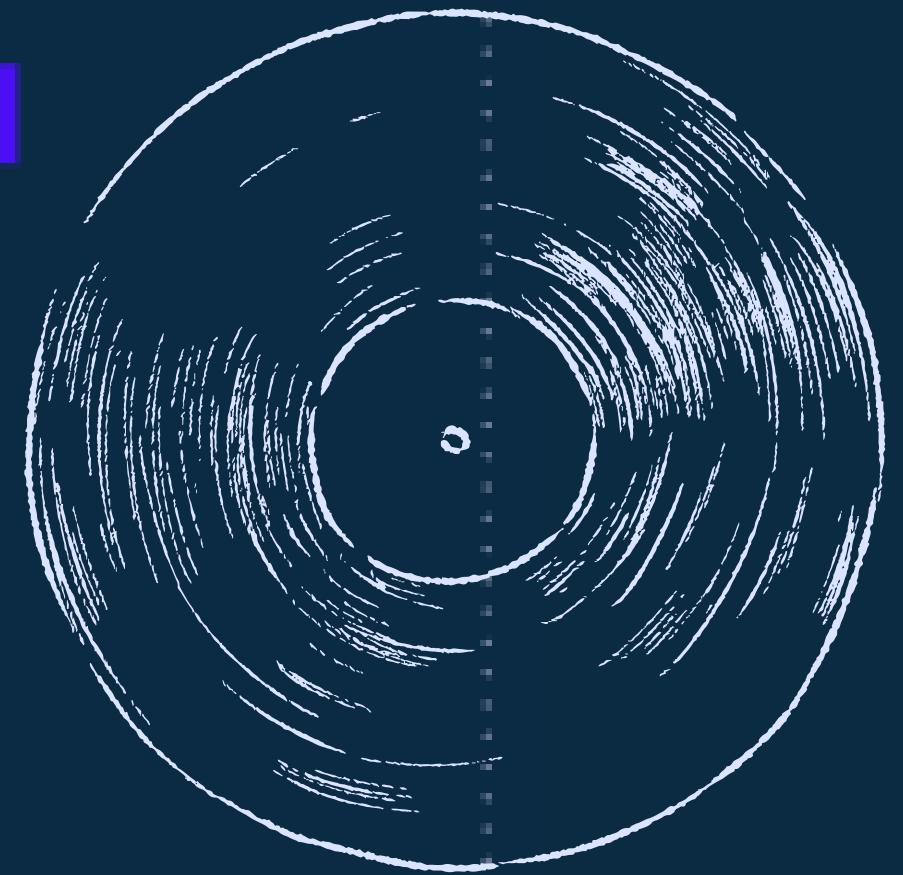
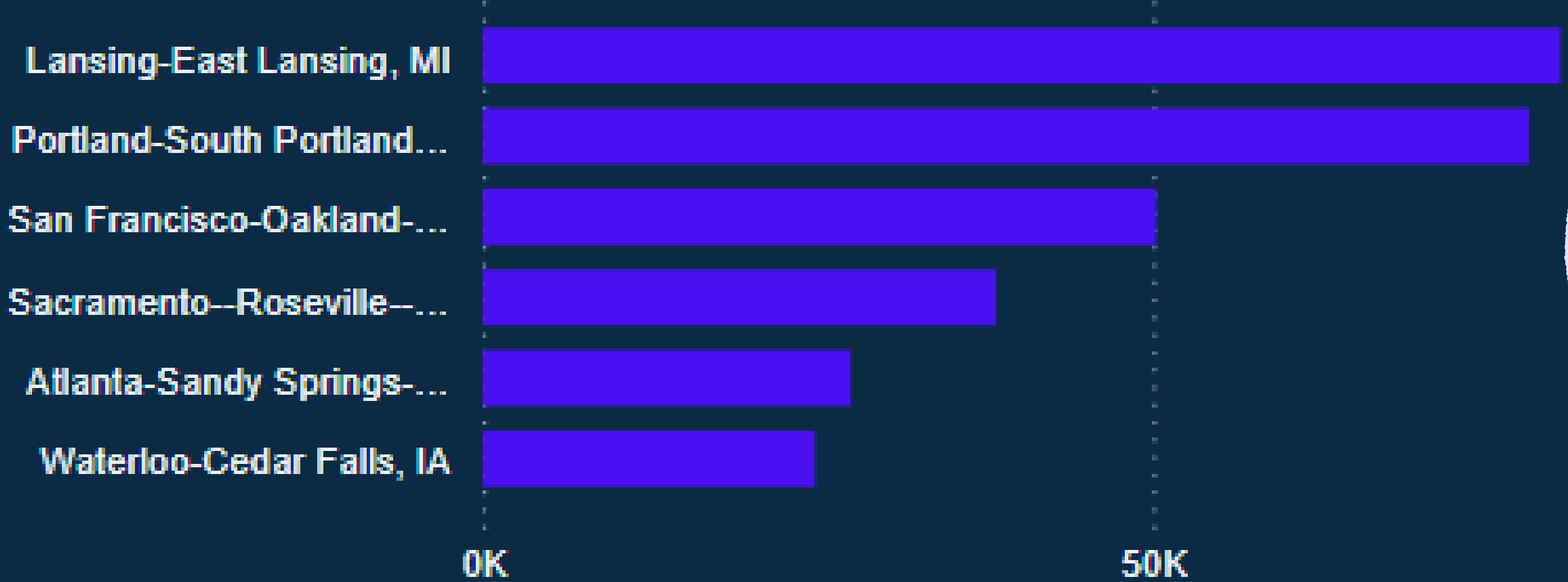


**Location**

- Atlanta-Sandy Springs-Roswell, GA
- Augusta-Richmond County, GA
- Birmingham-Hoover, AL
- Cedar Rapids, IA
- Chicago-Naperville-Elgin, IL-IN

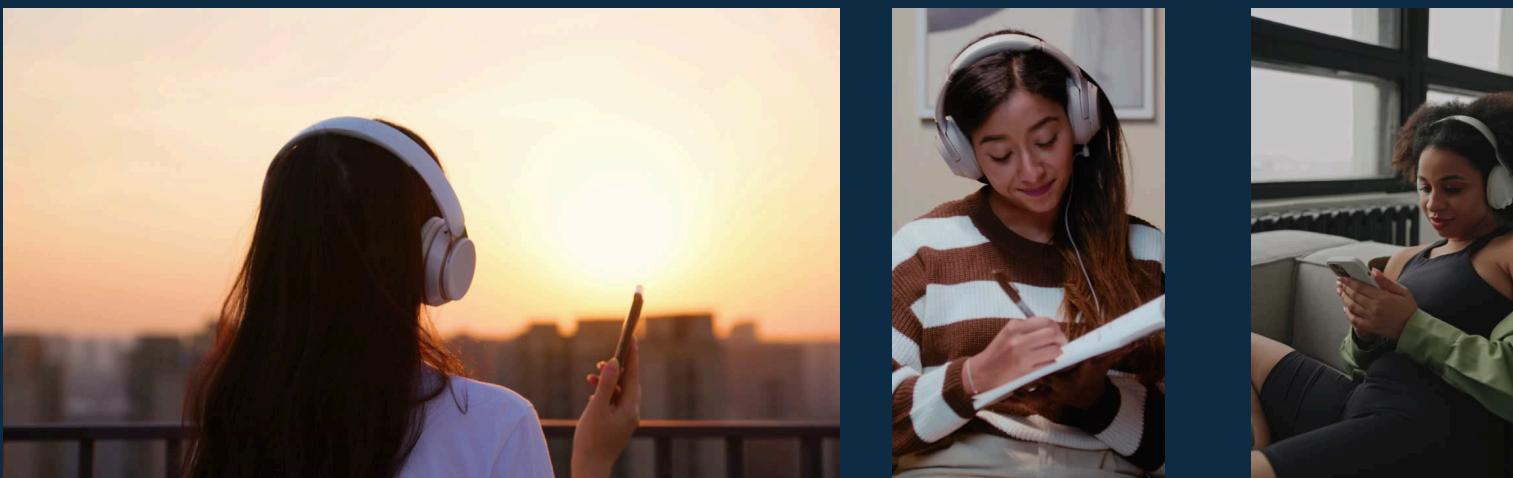
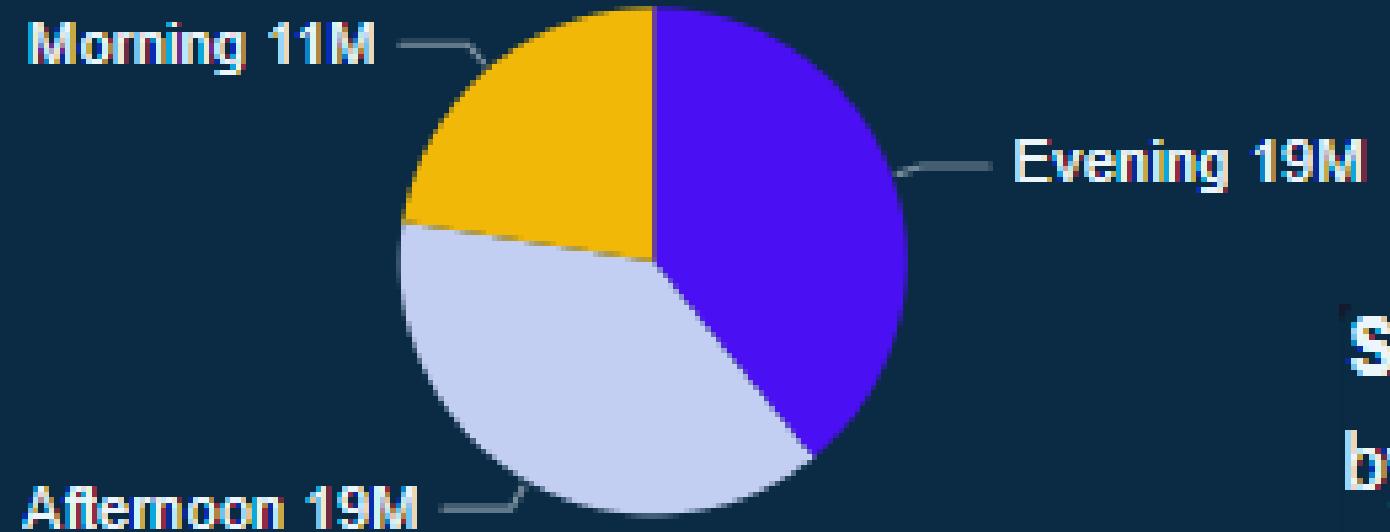


# Sum of user by location

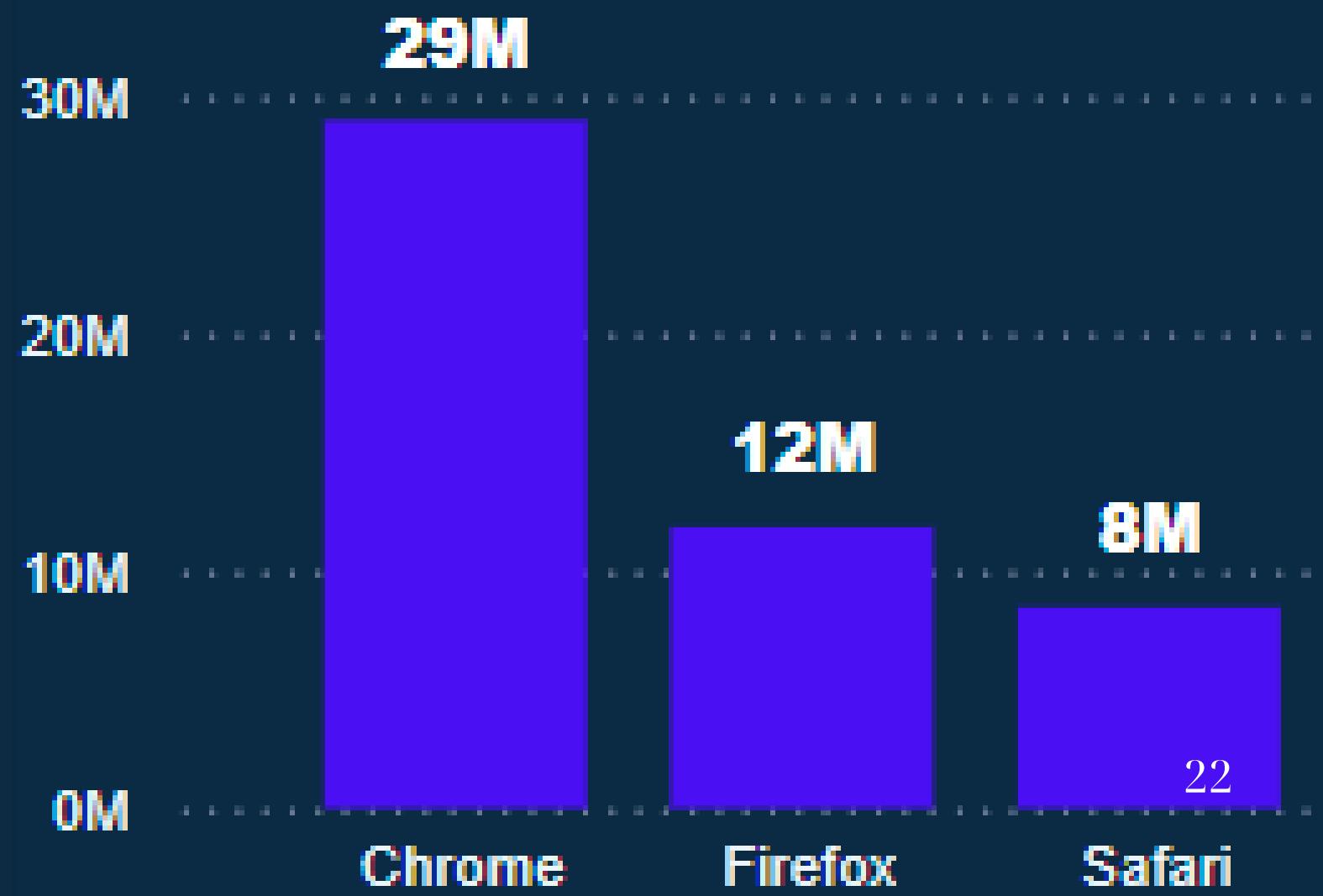


## Sum of songplay by time\_period

● Evening ● Afternoon ● Morning



## Sum of songplay by browser



# DATA MINING

## 1. DỰ ĐOÁN TRÌNH DUYỆT CỦA NGƯỜI DÙNG DỰA TRÊN HÀNH VI NGHE NHẠC

Bài toán: Xác định trình duyệt mà người dùng sẽ sử dụng để nghe nhạc dựa trên các đặc điểm như vị trí, thời gian nghe, nghệ sĩ yêu thích, độ dài bài hát, v.v.

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import LabelEncoder
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.tree import DecisionTreeClassifier
6 from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
7
8
9 df = df[['artist_name', 'location', 'browser']].copy()
10
11 # Mã hóa các cột
12 label_encoder = LabelEncoder()
13 df['artist_name'] = label_encoder.fit_transform(df['artist_name'])
14 df['location'] = label_encoder.fit_transform(df['location'])
15 df['browser'] = label_encoder.fit_transform(df['browser'])
16
17 # Tách dữ liệu thành input và output
18 X = df[['artist_name', 'location']]
19 y = df['browser']
20
21 # Chia tập dữ liệu thành tập train và test
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
1 # 2. Huấn luyện mô hình Decision Tree
2 model = DecisionTreeClassifier(
3     random_state=42,
4     max_depth=10, # Giới hạn độ sâu tối đa
5     min_samples_split=20, # Số mẫu tối thiểu để chia nút
6     min_samples_leaf=10, # Số mẫu tối thiểu tại mỗi lá
7 )
8 model.fit(X_train, y_train)
9
10 # 3. Tính Accuracy trên tập huấn luyện
11 y_train_pred = model.predict(X_train)
12 train_accuracy = accuracy_score(y_train, y_train_pred)
13 print("Train Accuracy:", train_accuracy)
14
15 # 4. Tính Accuracy trên tập kiểm tra
16 y_test_pred = model.predict(X_test)
17 test_accuracy = accuracy_score(y_test, y_test_pred)
18 print("Test Accuracy:", test_accuracy)
```

```
1 model = RandomForestClassifier(
2     random_state=42,
3     n_estimators=100, # Số lượng cây
4     max_depth=10, # Độ sâu tối đa của cây
5     min_samples_split=10, # Số mẫu tối thiểu để chia một nút
6     min_samples_leaf=5 # Số mẫu tối thiểu ở mỗi lá
7 )
8
9 model.fit(X_train, y_train) # Huấn luyện trên tập huấn luyện
10 # Dự đoán trên tập huấn luyện
11 y_train_pred = model.predict(X_train)
12
13 # Tính Accuracy trên tập huấn luyện
14 train_accuracy = accuracy_score(y_train, y_train_pred)
15 print("Train Accuracy:", train_accuracy)
16 # Dự đoán trên tập kiểm tra
17 y_test_pred = model.predict(X_test)
18
19 # Tính Accuracy trên tập kiểm tra
20 test_accuracy = accuracy_score(y_test, y_test_pred)
21 print("Test Accuracy:", test_accuracy)
22
```

# DATA MINING

Giải thuật	Train (%)	Test (%)	Train Accuracy (%)	Test Accuracy (%)
Decision Tree	80	20	94.39	94.28
Random Forest	80	20	93.32	93.14
Neural Network	70	30	80.10	80.29

- **Decision Tree và Random Forest thể hiện tính ổn định và có độ chính xác tương tự nhau, trong đó Decision Tree có độ chính xác cao hơn một chút.**
- **Neural Network có độ chính xác thấp hơn và có thể cần cải thiện thêm về cấu trúc cũng như các tham số huấn luyện.**

# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

- **Mục tiêu cuối cùng:** Tăng mức độ tương tác của người dùng với nền tảng âm nhạc, qua đó giúp tăng doanh thu. Giải pháp được đề xuất là xây dựng một hệ thống gợi ý bài hát, đề xuất top 10 bài hát có khả năng cao được người dùng nghe.
- **Collaborative Filtering**
  - **Matrix Factorization (SVD):** Phân rã ma trận người dùng - nghệ sĩ thành các ma trận tiềm ẩn để dự đoán mức độ yêu thích của người dùng đối với nghệ sĩ.
  - **Alternating Least Squares (ALS):** Cải tiến CF, tối ưu hóa ma trận bằng cách thay đổi các yếu tố người dùng và nghệ sĩ, phù hợp với dữ liệu thưa thớt và có khả năng mở rộng tốt.

# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

- **Content-Based Filtering**

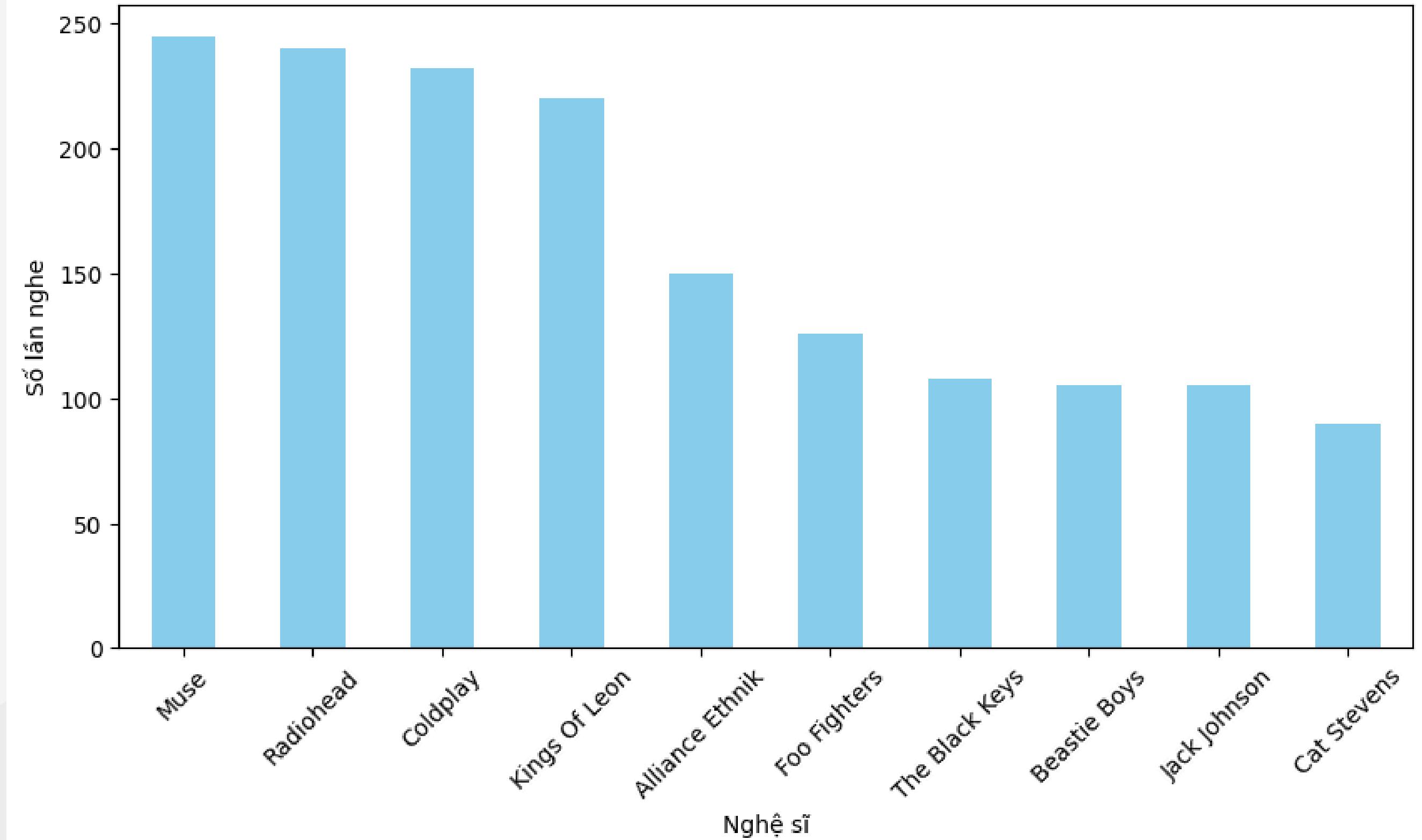
- Dựa vào đặc trưng bài hát và nghệ sĩ (như thể loại, phong cách âm nhạc) để tạo các gợi ý. Phân tích mô tả bài hát và nghệ sĩ (ví dụ, sử dụng TF-IDF hoặc embeddings như Word2Vec) để tính toán độ tương đồng giữa các nghệ sĩ.

- **Deep Learning**

- **Recurrent Neural Networks (RNN):** Dùng cho chuỗi dữ liệu theo thời gian (hành vi người dùng). RNN giúp học các xu hướng người dùng theo thời gian, chẳng hạn như người dùng có xu hướng nghe nhạc vào những thời điểm cụ thể trong ngày.
- **Transformer:** Mô hình mạnh mẽ để học các mối quan hệ lâu dài giữa các sự kiện người dùng, có thể nhận diện các thay đổi trong sở thích nghe nhạc theo thời gian.

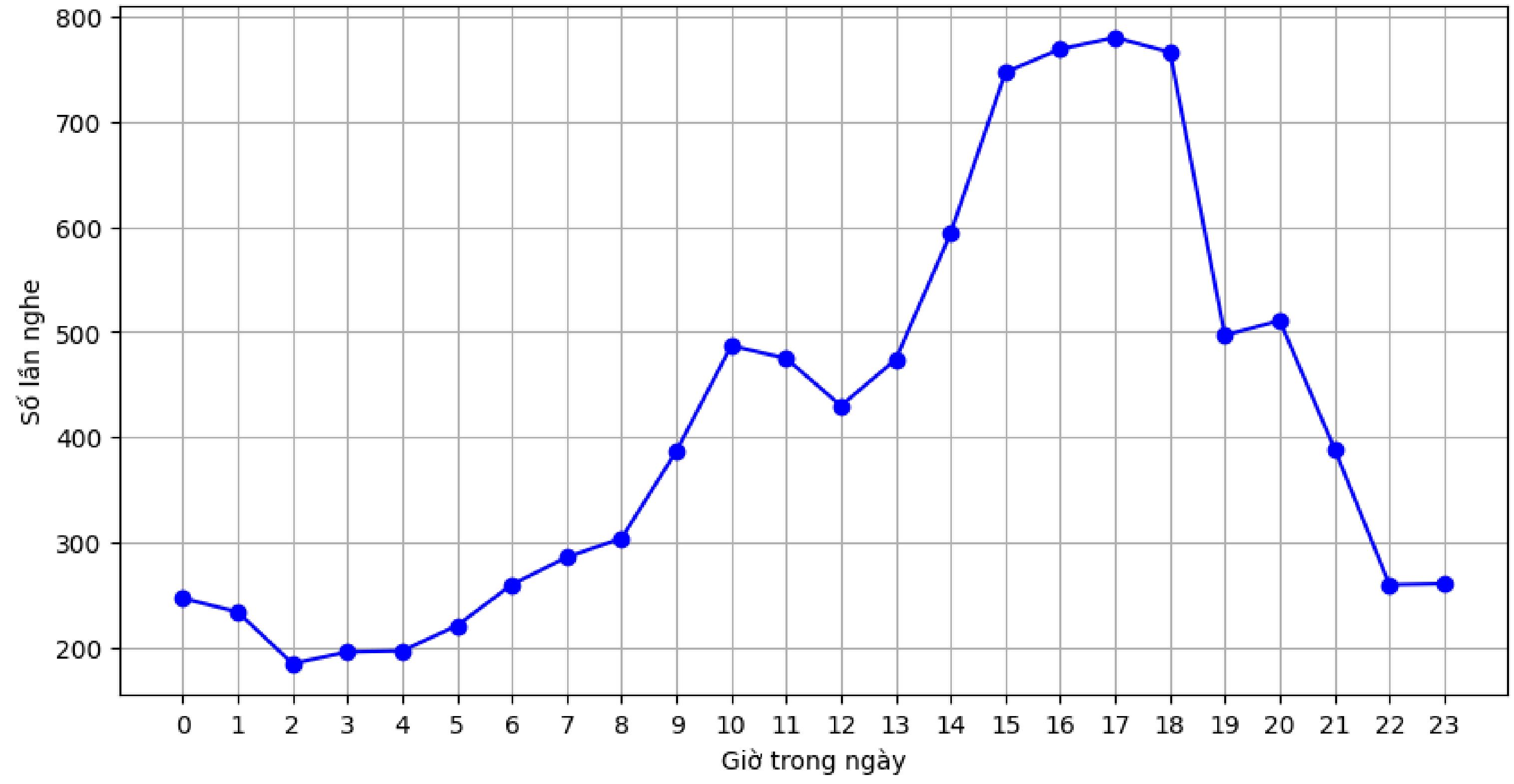
# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

Top 10 Nghệ Sĩ Được Nghe Nhiều Nhất

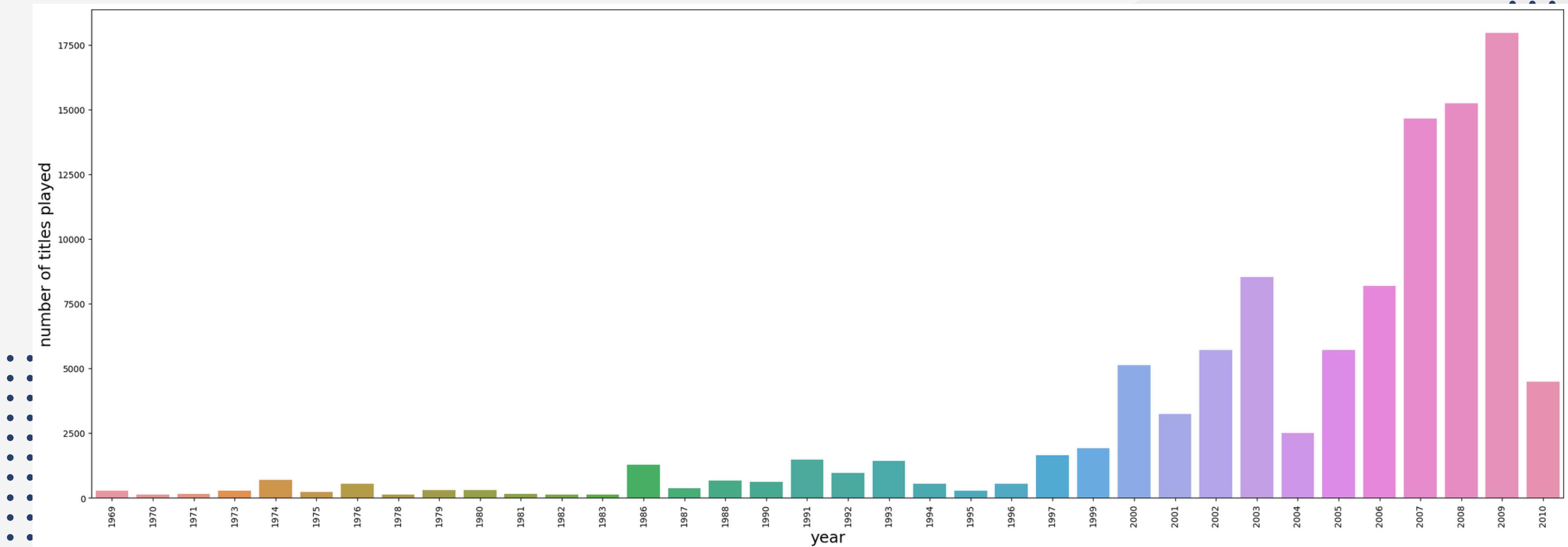


# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

Xu Hướng Nghe Nhạc Theo Giờ Trong Ngày



# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH



# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

## Quan sát về dữ liệu:

- Các bài hát phát hành từ năm 2000 trở đi phổ biến hơn, đặc biệt từ 2007-2009, nhờ sự phát triển công nghệ và thay đổi thói quen nghe nhạc. Các bài hát trước năm 2000 ít phổ biến hơn vì người dùng không phải khách hàng mục tiêu hoặc không ưa chuộng nhạc cũ. Sự giảm sút độ phổ biến của bài hát từ năm 2010 có thể do thiếu thời gian để người dùng nghe các bài hát mới.

## Giải pháp hướng đến:

- Hệ thống đề xuất âm nhạc dựa trên **mô hình item-item similarity-based (collaborative filtering)** có thể phù hợp hơn, do hành vi người dùng chủ yếu là nghe mỗi bài hát 1-2 lần. Dữ liệu hạn chế, đặc biệt là đối với các bài hát trước năm 2000, ảnh hưởng đến khả năng dự đoán sở thích người dùng.

# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

```
# Apply the best model found in the grid search
# Using the optimal similarity measure for item-item based collaborative filtering
sim_options = {'name': 'pearson_baseline',
               'user_based': False, 'min_support': 2}

# Creating an instance of KNNBasic with optimal hyperparameter values
sim_item_item_optimized = KNNBasic(sim_options = sim_options, k = 20, min_k = 6, random_state = 1, verbose = False)

# Training the algorithm on the train set
sim_item_item_optimized.fit(trainset)

# Let us compute precision@k and recall@k
precision_recall_at_k(sim_item_item_optimized)

...
RMSE: 0.3363

...
Precision: 1.0

...
Recall: 0.966

...
F_1 score: 0.983
```

# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

## Quan sát và thông tin chi tiết:

- Tính toán RMSE để kiểm tra mức độ sai lệch giữa số lượt phát dự đoán và số lượt phát thực tế. Trong trường hợp này là 0.3363.
- **Giải thích về Recall:** Mô hình đạt được recall ~0.966, có nghĩa là trong tất cả các bài hát phù hợp, 96.6% được đề xuất.
- **Giải thích về Precision:** Mô hình đạt được precision bằng 1, có nghĩa là trong tất cả các bài hát được đề xuất, 100% là các bài hát phù hợp.
- Điểm F1 của mô hình cơ bản là 0.983. Điều này có nghĩa là khoảng 98.3% bài hát được đề xuất là phù hợp và các bài hát phù hợp đã được gợi ý cho người dùng. Cải thiện mô hình này trong tương lai bằng cách sử dụng GridSearchCV để điều chỉnh các siêu tham số của thuật toán.

# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

```
# Applying the ranking_songs function
ranking_songs(recommendations, final_play)
```

	song_id	frequency_play	predicted_rating	corrected_rating
0	614	621	3.112439	3.072310
1	617	436	3.112439	3.064548
2	630	264	3.112439	3.050893
3	139	133	3.112439	3.025728
4	62	126	3.112439	3.023352

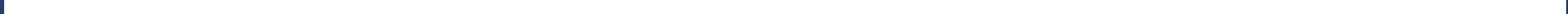
# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

		user_id	song_id	rating	text
		title			
	Learn To Fly	75901	1188	3.0	Learn To Fly There Is Nothing Left To Lose Foo...
	Everlong	9097	9249	3.0	Everlong The Colour And The Shape (Special Edi...
	The Pretender	49549	6525	3.0	The Pretender Echoes_Silence_Patience & Grac...
	Nothing Better (Album)	45386	1994	3.0	Nothing Better (Album) Give Up Postal Service
	From Left To Right	42302	4739	3.0	From Left To Right Corymb Boom Bip
	Lifespan Of A Fly	67704	3101	3.0	Lifespan Of A Fly Ray Guns Are Not Just The Fu...
...	Closer	8074	2527	3.0	Closer The Downward Spiral Nine Inch Nails
...	LDN	46525	7628	3.0	LDN LDN Lily Allen
...	Rianna	7320	5273	3.0	Rianna The Update Collection Vol. 2 Fisher
...	Eye Of The Tiger	7320	4399	3.0	Eye Of The Tiger Happy New Year! Survivor
...	What I've Done (Album Version)	7320	3744	3.0	What I've Done (Album Version) What I've Done ...

# DỰ ĐOÁN NGHỆ SĨ YÊU THÍCH

## So sánh các kỹ thuật và hiệu suất dựa trên chỉ số đánh giá:

- **Hiệu suất của các kỹ thuật khác nhau:** Hầu hết các mô hình đều xuất đạt kết quả tốt với F1 score lên tới ~0.98 sau khi tinh chỉnh siêu tham số.
- **Mô hình hoạt động tốt nhất:** Mô hình collaborative filtering dựa trên độ tương đồng giữa các item (item-item) là mô hình hoạt động tốt nhất.
- **Khả năng cải thiện thêm:** Do đa số điểm đánh giá là 3, cần thu thập thêm dữ liệu tương tác từ người dùng để hiểu rõ hơn về sở thích cá nhân. Nên xem xét thêm các chỉ số khác như MRR (Mean Reciprocal Rank) hay DCG (Discounted Cumulative Gain) để so sánh mô hình. Có thể giới hạn phạm vi dữ liệu từ năm 2000 trở đi để giảm nhiễu.



CẢM ƠN MỌI NGƯỜI  
ĐÃ LẮNG NGHE!

